AccelStack: A Cost-Driven Analysis of 3D-Stacked LLM Accelerators

Chen Bai[1,†,‡], Xin Fan[1,†], Zhenhua Zhu[1,2], Wei Zhang[1], Yuan Xie[1]
[1] The Hong Kong University of Science and Technology [2] Tsinghua University

***Abstract*—Large language models (LLMs) show viability for artificial general intelligence (AGI) with high computing power and memory bandwidth demands. While existing LLM accelerators leverage high-bandwidth memory (HBM) and 2.5D packaging to address the challenge, emerging hybrid bonding techniques unlock new opportunities for 3D-stacked LLM accelerators. This paper proposes AccelStack, a cost-driven analysis for the new architecture via two innovations. First, a performance model capturing memory-on-logic is presented. Second, a cost model for die-on-die (DoD), die-on-wafer (DoW), and wafer-on-wafer (WoW) is proposed. Evaluations show 3D-stacked accelerators achieve up to** $7.17\times$ **and** $2.09\times$ **faster inference than NVIDIA A100 (FP16) and H100 (FP8) simulation results across various LLM workloads, with chiplet-based designs reducing recurring engineering costs by** 38.09% **versus monolithic implementations.**

I. Introduction

Large language models (LLMs) are profoundly changing our industry, driving innovations from code generation, intelligent agents, to complex problem-solving with a slow-thinking fashion [1]–[3].

The inference process of LLMs can be divided into two key stages: prefill and decoding. The prefill stage is compute-bound, primarily due to its reliance on general matrix multiplications (GEMM). In contrast, the decoding stage is dominated by general matrix-vector multiplications (GEMV), making it inherently memory-bound. Beyond raw computing power, this auto-regressive inference process places substantial demands on memory bandwidth.

Many LLM accelerators are proposed to catch up with high computing power and memory bandwidth demand accordingly [4]–[9]. The main ideas behind existing solutions are threefold. *First, customized computing units are applied.* Systolic array, tensor cores, 3D cube unit, *etc.*, are proposed to handle the massive GEMM operations [4]–[7]. *Second, architect LLM accelerators with new memory technology.* HBM is adopted to replace traditional double-data-rate (DDR) SDRAMs [4]–[6], [9]. By vertically integrating multiple SDRAM dies using microbumps, HBMs achieve far wider data buses, resulting in significant memory bandwidth improvements [10]. *Third, advanced 2.5D packaging brings memory closer to the compute units, leading to reduced memory access latency.* For example, silicon interposers or bridges substitute conventional dual in-line memory module (DIMM) links with shorter interconnections [11], [12].

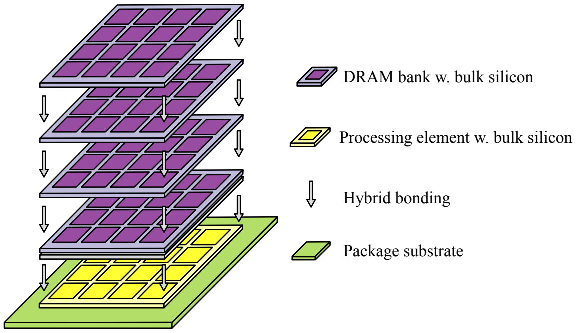


Fig. 1 Overview of 3D-stacked LLM accelerator architecture landscape. Multiple DRAM dies are hybrid-bonded together and subsequently hybrid-bonded to the underlying compute die. The design is then mounted onto a package substrate.

New opportunities are now emerging with advancements in hybrid bonding techniques [13], [14]. The advancements make 3D-stacked architectures a compelling solution for next-generation LLM accelerators by enabling denser and shorter vertical interconnects compared to existing solutions. So, we propose a landscape of 3D-stacked LLM accelerator architecture in this paper accordingly (Fig. 1). We consider the memory-on-logic design rather than logic-on-memory, as the latter requires additional space for through-silicon vias (TSVs) to deliver power to the logic die, increasing die size and reducing memory density [15]. The proposed architecture landscape offers two immediate benefits. First, bonding pitches and pad sizes are $5 \sim 30\times$ smaller than those of microbumps [14]. Hence, higher memory bandwidth than HBM is available. Second, compared to 2.5D packaging, the shorter interconnect distances between memory and compute units indicate that smaller gate-to-gate delay and lower power dissipation per bit transmission are free lunches.

However, these opportunities come with notable challenges. The massive adoption of 3D-stacked LLM accelerators hinges on two critical technological metrics: performance and manufacturing cost. Performance quantifies its superiority over traditional LLM accelerators, whereas cost determines its economic feasibility. Ensuring cost-effectiveness remains a primary concern for investments in high-volume manufacturing (HVM) of 3D-stacked LLM accelerators.

To address this gap, we present AccelStack, a cost-driven analysis framework for 3D-stacked LLM accelerators. AccelStack integrates performance analysis and cost modeling

† Both authors contributed equally to this work.
‡ Corresponding author.

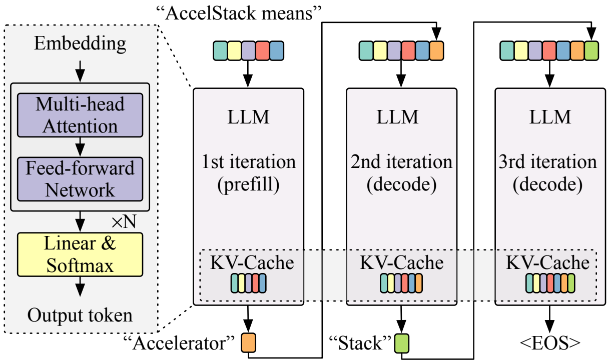


Fig. 2 Overview of LLM inference.

to evaluate the feasibility of the architecture. Its highlights span two aspects. First, AccelStack introduces a holistic performance model that captures achievable memory bandwidth improvements due to hybrid bonding, while also supporting "chipletized" designs with diverse packaging techniques and distinct LLM parallelization strategies. Second, a cost model is presented. Especially, it involves the modeling for different hybrid-bonding-enabled 3D integration manufacturing flows, including die-on-die (DoD), die-on-wafer (DoW), and wafer-on-wafer (WoW) [1]. We aim to identify the new architecture's performance-cost trade-offs, paving the way for the practical deployment in real-world applications. Our contributions are summarized as follows:

- We propose a landscape of 3D-stacked LLM accelerators, along with their chipletized variants.
- A performance model that characterizes 3D-stacked LLM accelerators is introduced.
- A cost model supports DoD, DoW, and WoW hybrid bonding manufacturing flows is proposed in AccelStack.
- Evaluations show 3D-stacked accelerators achieve up to 7.17× and 2.09× faster inference than A100 (FP16) and H100 (FP8) simulation results across LLM workloads, with chiplet-based designs reducing recurring engineering costs by 38.09% versus monolithic implementations.

The remainder of this paper is organized as follows. Section II provides preliminaries. Section III details AccelStack framework. Section IV is for experiments. Finally, Section V concludes this paper.

II. Preliminaries

In this section, Section II-A overviews LLM inference, Section II-B summarizes 3D integration and chiplet technology, and Section II-C introduces cost analysis.

A. LLM Inference

LLM inference is autoregressive, generating each token based on previously generated ones. The prefill stage generates the first output token, while the decoding stage produces the subsequent tokens until the response is complete.

Fig. 2 shows an inference pipeline. The example input sequence (*e.g.*, "AccelStack means") is denoted by an array of tokens with different colors for better visualization. The token sequence is handled by stacked layers of multi-head attention (MHA) operation and feed-forward network. MHA is defined as

$$\mathrm{MHA}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \mathtt{softmax}(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d}})\boldsymbol{V}, \qquad (1)$$

where $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ are query, key, and value matrices, respectively. d is the dimension of the query or key in Equation (1).

During the prefill stage, the KV cache is generated to accelerate the inference [17]. The first output token (*i.e.*, "Accelerator") is appended to the previous input tokens after the prefill stage. In the decode stage, tokens are reprocessed by the LLM, generating a new token in the next iteration, until a special token (*e.g.*, "<EOS>") is generated. The special token signifies a finished inference. In each iteration, the KV cache is reused and enlarged due to new tokens generated.

B. 3D Integration & Chiplet Technology

3D integration stacks multiple chips vertically to create a higher-density integrated circuits. TSVs are needed. Two mainstream approaches are widely adopted. The first uses microbumps, which are small, raised metal structures, with diameters in the tens of micrometers. HBM and Ponte Vecchio are examples [10], [18]. The second is bumpless integration with hybrid bonding [14]. Hybrid bonding combines dielectric and metal bonding (*e.g.*, Cu-Cu bond [19]). This process begins with low-temperature and low-pressure alignment, followed by annealing at $150 \sim 300$°C [20].

Chiplet technology reduces manufacturing costs at advanced technology nodes by disassembling a large monolithic system-on-chip (SoC) into several smaller chips [18], [21], [22]. These smaller chips are interconnected and packaged together to perform the same functions as the original monolithic design. Besides lowering manufacturing costs, chiplet technology offers two additional merits. On one hand, die-to-die interfaces and interconnections, *e.g.*, universal chiplet interconnect express (UCIe), facilitates chiplet reuse [23]. This allows the production of new, low-cost products while meeting strict time-to-market requirements. On the other hand, it enables hetergeneous integration. Specifically, multiple chiplets manufactured using different technology nodes, materials, and sourced from various fabless design houses and foundries can be packaged together. The packaging solutions encompass conventional multi-chip module (MCM) and advanced 2.5D packaging, such as CoWoS (Chip-on-Wafer-on-Substrate) and EMIB (Embedded Multi-Die Interconnect Bridge). [11], [12], [24]. MCM integrates dies onto a unified package substrate. CoWoS with silicon interposer (CoWoS-S) employs a full-sized interposer, while EMIB utilizes localized bridge structures [11], [12].

C. Cost Analysis

The cost of 3D-stacked LLM accelerators can be divided into recurring engineering (RE) costs and non-recurring engineering (NRE) costs [25]. NRE costs are one-time expenses incurred during the design phase, covering activities such as software

[1] Although hybrid-bonding in the DoD flow is not yet ready for volume production, AccelStack supports this technique in anticipation of its eventual realization [16].

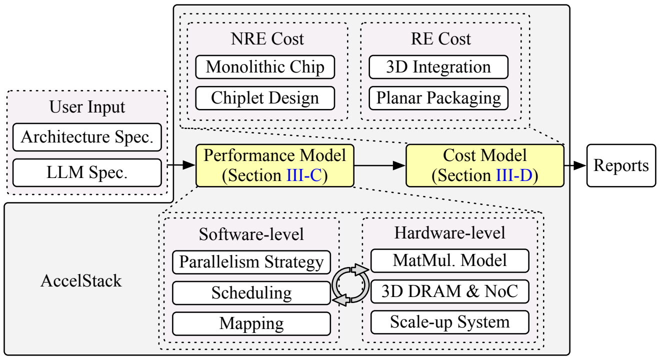


Fig. 3 Overview of AccelStack.

tools, IP licensing, chiplet/module/package design, verification processes, and mask production. These costs are independent of production volume and represent the disposable investment required to develop the chip. Oppositely, RE costs are tied to mass production, including wafer fabrication, packaging, and post-production testing.

The total engineering cost combines RE costs and the amortized portion of NRE costs, which is distributed across the production volume. Amortization is influenced by production scale, *i.e.*, in low-volume production, NRE costs dominate the overall expense, whereas for HVM, RE costs prevail as NRE's per-unit share diminishes

III. AccelStack

This section begins with an overview of the framework in Section III-A, followed by the modeled architecture design in Section III-B. Section III-C describes the performance model, while Section III-D presents the cost model.

A. Overview of AccelStack

AccelStack integrates two core models: a performance model and a cost model. Given architecture and LLM workload specifications, AccelStack conducts both performance and cost analysis (Fig. 3). Architecture specifications include microarchitecture parameter settings such as link bandwidth, computing power, packaging techniques, *etc.* LLM workload specifications cover the LLM model structure, input/output sequence lengths, and batch size.

The performance analysis consists software and hardware levels. At the software level, AccelStack generates a design space for LLM inference parallelization strategies, along with related scheduling and mapping mechanisms. The parallelization strategies involve pipeline parallelism (PP), data parallelism (DP), tensor parallelism (TP) with sequence parallelism (SP), and expert parallelism (EP) [17], [26]. At the hardware level, internal models are used to model hardware executions, such as performing matrix multiplication, network-on-chip (NoC) transmission, *etc.* The performance analysis is iterative, continuing until all viable parallelization strategies are evaluated or a near-optimal candidate is identified within a predefined time budget.

After completing performance analysis, AccelStack evaluates costs, including NRE and RE. The NRE cost model supports both monolithic and chiplet-based designs, while the RE cost model accounts for 3D integration and planar packaging. Notably, MCM, CoWoS, and EMIB are termed as planar packaging in AccelStack as these technologies integrate chips in horizontal directions [11], [12], [24]. Finally, AccelStack generates a quality-of-results (QoR) report.

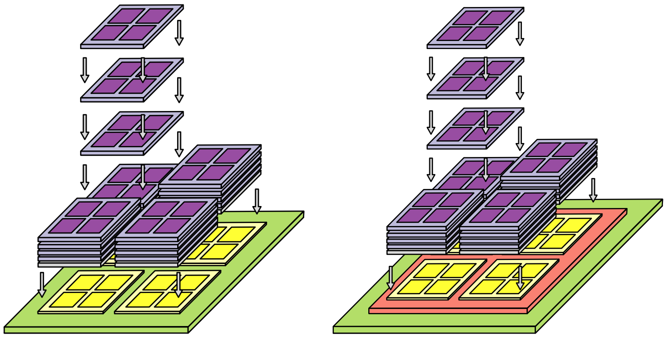

(a) 3D chiplet design (b) 3.5D chiplet design

Fig. 4 Chipletized versions of the architecture are designed based on Fig. 1. In Fig. 4(b), the plane colored in red denotes a silicon interposer.

B. Architecture Design

The architecture landscape modeled by AccelStack is depicted in Fig. 1, representing a monolithic design. From top to bottom, the architecture involves multiple DRAM dies, a computing die, and a package substrate. Hybrid bonding is used to connect neighboring DRAM dies as well as to link DRAM dies to the computing die. A 2D array of processing elements (PEs) resides on the computing die, aligned with the DRAM channels above it. The chip stack is then mounted onto the package substrate. The areas of both the DRAM dies and the computing die approach the reticle limit, serving as our strawman solution for integrating memories and PEs with hybrid bonding. For simplicity, details such as NoC is omitted from the visualization.

Based on Fig. 1, we propose chipletized versions of the architecture correspondingly (Fig. 4). The monolithic design is disassembled into four 3D chiplets (Fig. 4(a) and Fig. 4(b)). Depending on the presence of interposers, these architectures are referred to as 3D chiplet design and 3.5D chiplet design. The term "3.5D" arises from the simultaneous adoption of both 3D integration and 2.5D packaging techniques. Each 3D chiplet contains a subset of the DRAM dies and the computing die extracted from Fig. 1. Moreover, die-to-die interconnections implementing UCIe are necessary to achieve functional equivalence with the original monolithic design. Furthermore, the chipletized design can be packaged using alternative techniques to achieve different cost-performance trade-offs. Fig. 4(a) employs a typical MCM, whereas Fig. 4(b) uses CoWoS [11], [24]. Notably, EMIB can also be applied for 3.5D chiplet design [12]. The use of silicon interposers or bridges can significantly enhance chiplet-to-chiplet bandwidth compared to MCM, albeit at the expense of higher manufacturing costs.

C. Performance Model

We demonstrate the performance model using 3D chiplet design (Fig. 4(a)). Fig. 5 shows the microarchitecture. In the

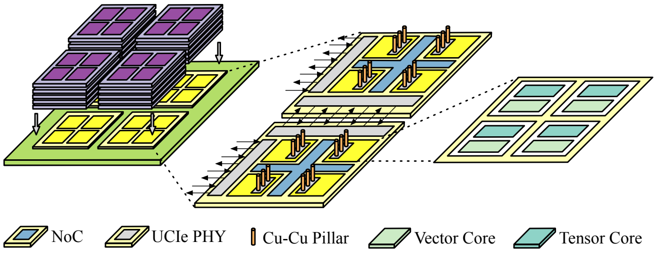


Fig. 5 AccelStack's modeled microarchitecture for Fig. 4(a).

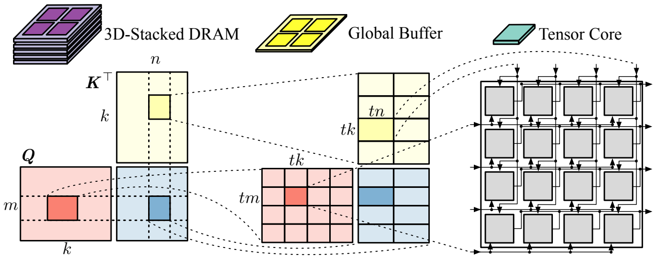


Fig. 6 The two-level loop tiling of the GEMM operation.

computing die, four chiplets are interconnected via UCIe, with UCIe PHYs positioned along the appropriate chiplet shorelines [23]. Serializer/Deserializer (SerDes) circuits are integrated into each chiplet, enabling the entire chip to scale-up with other chips, similar to NVLink-based systems (not shown in Fig. 5) [6]. Each chiplet has four PEs, and every PE has four cores. Inside a core, hybrid bonding pillars (*viz.*, Cu-Cu pillars) connect a DRAM controller and DRAM channels located above it. Besides, each core incorporates buffers, registers, and modules like vector core, tensor core, and other functional units for non-linear mathematics. In contrast, UCIe PHYs are not required for the monolithic design (Fig. 1). Therefore, all PEs are communicated through a single NoC.

In the following, we introduce the modeling methods of matrix multiplication (Section III-C1), 3D hybrid-bonded DRAM (Section III-C2), NoC communication (Section III-C3), and scale-up system (Section III-C4).

1) Matrix Multiplication

Given a GEMM operation like $\boldsymbol{QK}^\top$ shown in Equation (1), AccelStack partitions $\boldsymbol{Q}$ and $\boldsymbol{K}^\top$ *w.r.t.* each chiplet and models the sub-matrix multiplication using the two-level loop tiling (Fig. 6). For example, the partitioned $\boldsymbol{Q}$ has dimensions $m \times k$, while the partitioned $\boldsymbol{K}^\top$ is with $k \times n$ for each chiplet. In the first level tiling, the two matrices are further divided into submatrices. These submatrices are transferred between DRAM and the global buffer (located on the computing die) on a per-subtile basis. In the second level of tiling, the PE continues to partition the submatrices into smaller tiles with dimensions $tm \times tk$ and $tk \times tn$. The tensor core accepts inputs of a predetermined shape and outputs the results. If the input matrices do not match the required dimensions of the tensor core, preprocessing steps such as zero padding are applied. Dashed lines in Fig. 6 show the related data flow. So, the pure computing latency can be characterized by Equation (2).

$$\begin{aligned} t_{\text{comp}} = & \frac{m+tm-1}{tm} \cdot \frac{n+tn-1}{tn} \cdot \frac{k+tk-1}{tk} \\ & \cdot \frac{2 \cdot (tm+p-1) \cdot (tn+p-1) \cdot tk}{p^2 \cdot \alpha \cdot P}, \end{aligned} \tag{2}$$

where the tensor core is a $p \times p$ array of multiplier-accumulator units, P denotes the number of outputs per cycle, α is the utilization, and 2 arises from counting both multiplication and addition in each FMA operation [2].

2) 3D Hybrid-Bonded DRAM

Unlike HBM, 3D hybrid-bonded DRAM features distributed controllers and PHYs, which are aligned with PEs. While canonical HBM access can incur latencies exceeding 300 ns, AccelStack estimates the access latency of 3D hybrid-bonded DRAM as over 2× faster than that of HBM due to shorter vertical interconnects [27]. The timing parameters are validated against vendor data [28]. More details are provided in Section IV-A1.

3) NoC Communication

Link model is used in NoC communication [29]. Consider a scenario where a link needs to transfer n bytes of data from the source to the target. Let the payload size and flit size used by the NoC transmission protocol be denoted as `payload` and `flit`, respectively. Equation (3) specifies the NoC transfer latency accordingly.

$$t_{\text{comm}} = \frac{V}{B} + l + o, \quad V = \left\lceil \frac{n}{\texttt{payload}} \right\rceil \cdot \texttt{flit}, \tag{3}$$

where B is the link bandwidth, l is the fixed latency of the physical layer, and o denotes the additional overhead.

4) Scale-Up Systems

As the scale of LLMs continues to grow from billions to trillions of parameters, inference must be distributed across multiple chips. Scale-up system is a distributed inference system with multiple LLM accelerators and network switches. It is constructed by enabling chip-level peer-to-peer accesses with load-store semantics, allowing a single system to behave as if it were one giant chip. The system consists of multiple "nodes", with each node containing a collection of chips and network switches. AccelStack models scale-up systems using Clos-based interconnects, a classical data center network architecture that has been adopted in commercial super nodes. [30], [31]. Particularly, AccelStack reuses Equation (3) for inter-rack communication.

D. Cost Model

The main idea of the cost model in AccelStack is to compute each component of cost separately and then sum them up, as expressed in Equation (4):

$$C = C_{\text{NRE}} + C_{\text{RE}}, \tag{4}$$

where C_{NRE} and C_{RE} are for NRE and RE cost, respectively. We first introduce the NRE cost modeling, followed by the RE cost modeling.

As discussed in Section II-C, NRE costs are one-time expenses. Modules like tensor cores, DRAM, NoC, *etc.*, are

[2] Part of memory accesses cannot be overlapped with the computation, however, we omit the illustration due to the page limit.

designed in parallel and subsequently integrated to form the monolithic design (Fig. 1) within a chip-integrated product development (IPD) flow [32]. These costs are challenging to estimate because they depend on numerous factors, many of which such as design complexity and the number of person-months invested, are case-specific. To address this, we develop an NRE cost model using a first-order approximation, *i.e.*, leveraging silicon area as a cost metric, following Feng *et al.* [25]. The NRE cost is proportional to the silicon area of each module (Equation (5)).

$$C_{\text{NRE}} = (\sum_{m_i \in \text{chip}} \alpha_{m_i} A_{m_i}) + \beta_{\text{chip}} A_{\text{chip}} + C_{\text{fixed}}, \tag{5}$$

where chip represents the monolithic design, A_{m_i} is the silicon area of the i-th module, and α_{m_i} serves as its corresponding cost coefficient. Similarly, A_{chip} is the total silicon area of the monolithic design. And β_{chip} accounts for physical design and verification efforts. C_{fixed} refers to the requisite fixed costs, involving EDA tools purchases, IP licensing, and full masks sets.

The NRE cost of the chiplet design (Fig. 4) is described in Equation (6):

$$C_{\text{NRE}} = \sum_{m_i \in M} \alpha_{m_i} A_{m_i} + (\sum_{c \in \text{chip}} \beta_c A_c + C_{\text{fixed}_c}) + C_{\text{fixed}}, \tag{6}$$

where M represents the set of modules. A_c, β_c, and C_{fixed_c} denote the area, cost coefficient, and per-chip fixed cost, respectively, for the c-th chiplet of the design. The philosophy behind the chiplet design's ability to reduce NRE costs lies in the design reuse of modules. This is reflected in the comparison of the first term between Equation (5) and Equation (6). Namely, the monolithic design typically includes a much larger number of modules contributing to the NRE cost.

The RE cost is the repetitive expenses incurred during the mass production stage. It consists of three components: logic die production, DRAM die production, and packaging. The RE costs for a logic die and a DRAM die are similar. Taking the logic die as an example, we use the negative binomial model to compute the logic die yield (Equation (7)).

$$Y_{\text{logic}} = Y_{\text{wafer}} \cdot (1 + \frac{A_{\text{logic}} D_0}{\alpha})^{-\alpha}, \tag{7}$$

where Y_{wafer} denotes the wafer yield, A_{logic} represents the silicon area of the logic die, and D_0 is the defect density, which correlates with the technology node. The number of rectangular logic dies per circular wafer is approximated by Equation (8) [33].

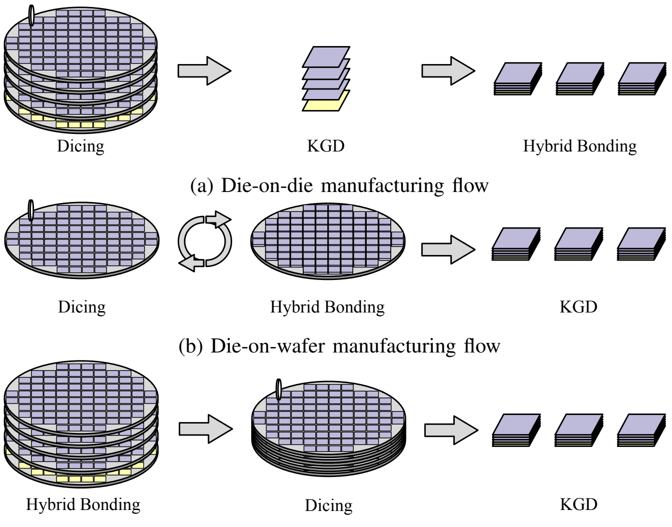
$$N_{\text{logic}} = \frac{\pi \cdot (\frac{\phi_{\text{wafer}}}{2})^2}{A_{\text{logic}}} - \frac{\pi \cdot \phi_{\text{wafer}}}{\sqrt{2 \times A_{\text{logic}}}}, \tag{8}$$

where ϕ_{wafer} is the diameter of a circular wafer. Thus, the RE cost per logic die can be written as follows:

$$C_{\text{logic}} = (\frac{C_{\text{wafer}}}{N_{\text{logic}}} + C_{\text{test}}) / Y_{\text{logic}}. \tag{9}$$

In Equation (9), C_{test} is the known good die (KGD) test cost.

The RE cost of packaging includes the hybrid bonding process and planar packaging (Section III-A). In the following paragraphs, we first present the cost modeling for the hybrid bonding process and then detail the planar packaging.



(a) Die-on-die manufacturing flow

(b) Die-on-wafer manufacturing flow

(c) Wafer-on-wafer manufacturing flow

Fig. 7 Overview of the DoD, DoW, and WoW hybrid bonding process manufacturing flows. DRAM dies are highlighted in purple, while logic dies are highlighted in yellow.

The hybrid bonding process features three distinct manufacturing flows: die-on-die (DoD), die-on-wafer (DoW), and wafer-on-wafer (WoW), as summarized in Fig. 7. The three flows achieve different trade-offs between production efficiency, bonding density and cost. We examine each of these flows in detail below.

1) *Die-on-die*

DoD indicates that DRAM chips and logic chips are hybrid-bonded at the die granularity (Fig. 7(a)). Suppose n dies need to be hybrid-bonded using this fashion. For a single product, let C_{DoD} denote the cost per bonding process, and Y_{DoD} represent the yield of each process. The overall cost for each design is computed using Equation (10):

$$C_{\text{3D}} = \frac{\sum_{i=1}^{n} C_{\text{die}_i} + (n-1) C_{\text{DoD}}}{\prod_{i=1}^{n-1} Y_{\text{DoD}}}, \tag{10}$$

where

$$C_{\text{die}_i} = (\frac{C_{\text{wafer}}}{N_{\text{die}}} + C_{\text{test}} + C_{\text{misc.}}) / Y_{\text{die}}, \tag{11}$$

and $C_{\text{misc.}}$ is the additional costs associated with processes like die thinning and alignment for hybrid bonding. We adapt Equation (9) with minor revisions to obtain Equation (11).

2) *Die-on-wafer*

The manufacturing flow of DoW is illustrated in Fig. 7(b). First, a wafer is diced into individual dies, followed by hybrid bonding between the selected good die and another wafer. These two steps are repeated multiple times until the chip stacking process is complete. Similarly, C_{DoW} and Y_{DoW} are the cost and yield of DoW, respectively. As a result, the overall cost for each design is manifested below (Equation (12)).

$$
\begin{aligned}
C_{\text{3D}} &= (\frac{C_{\text{logic wafer}} + C_{\text{DoW}}}{N_{\text{logic}}} + C_{\text{DRAM}_{i+1}} + C')/(Y_{\text{logic}} \cdot Y_{\text{DoW}}), \\
C_{\text{DRAM}_{i+1}} &= (\frac{C_{\text{DRAM wafer}} + C_{\text{DRAM}_i} + C_{\text{DoW}}}{N_{\text{DRAM}}} + C')/Y_{\text{DoW}},
\end{aligned}
\tag{12}
$$

where $C' = C_{\text{test}} + C_{\text{misc.}}$, $i = 0, 1, ..., n-2$, C_{DRAM_0} is from Equation (11), and other symbols with subscripts best explain their own meanings. It is worth noting that N_{logic} and N_{DRAM} are identical since the hybrid bonding process requires the silicon areas of logic and DRAM dies to be close for precise die alignment.

3) Wafer-on-wafer

WoW directly hybrid-bonds multiple DRAM wafers to the logic wafer, as displayed in Fig. 7(c). The overall cost per design is then calculated using:

$$
\begin{aligned}
C_{\text{3D}} &= (\frac{\sum_{i=1}^{n} C_{\text{wafer}_i} + (n-1) \cdot C_{\text{WoW}}}{\min\{N_{\text{logic}}, N_{\text{DRAM}}\}} + C')/Y' \\
Y' &= Y_{\text{logic}} \cdot Y_{\text{DRAM}} \cdot \prod_{i=1}^{n-1} Y_{\text{WoW}}.
\end{aligned}
\tag{13}
$$

In Equation (13), we reuse symbols discussed earlier. Additionally, $\prod_{i=1}^{n-1} Y_{\text{DRAM}}$ is not applied because defects are predominantly systematic and exhibit negligible randomness.

The most notable difference regarding cost among DoD, DoW, and WoW lies in the bonding and testing expenses. In other words, DoW requires higher alignment precision compared to WoW, while DoD is not yet ready for mass production [16].

The 2.5D packaging modeled by AccelStack is show in Equation (14).

$$
\begin{aligned}
C_{\text{package}} = &C_{\text{raw package}} + \\
&C_{\text{interposer}} \cdot (\frac{1}{Y_1 \cdot Y_2^n \cdot Y_3} - 1) + \\
&C_{\text{substrate}} \cdot (\frac{1}{Y_3} - 1) + \\
&C_{\text{3D}} \cdot (\frac{1}{Y_2^n \cdot Y_3} - 1),
\end{aligned}
\tag{14}
$$

where Y_1 is the yield of the interposer or bridge structure, Y_2 is the bonding yield of chips, Y_3 is the bonding yield of the interposer or bridge structure, and C_{3D} is computed from Equation (10), Equation (12), or Equation (13), depending on the selected hybrid bonding manufacturing flow. Since the interposer or bridge used in CoWoS or EMIB is manufactured using silicon, its cost can be computed similarly to Equation (9). The yield of the interposer and the number of interposers produced per wafer can be referenced from Equation (7) and Equation (8) [3]. For traditional MCM packaging, no interposer is required. Therefore, we exclude the term involving $C_{\text{interposer}}$ and let $Y_3 = 1$ in Equation (14).

TABLE I The LLMs used for experiments.

Model	Parameters	Vocab size	Context window	Layers	Attention [1]
Llama3	8B/70B/405B	125K	8K	126	MHA & GQA
Gemma2	2B	250K	8K	26	GQA
Qwen2	72B	148K	32K	80	GQA
DeepSeek-v3	671B	126K	128K	61	MLA

[1] GPA and MLA are short for group query attention and multi-latent attention, respectively.

In summary, the RE cost C_{RE}, can be characterized by C_{package} from Equation (14). And with Equation (4), we can evaluate the total cost of a design.

IV. Experiments

This section presents the experiments. First, we introduce experimental methodology in Section IV-A. Then, we provide results and analysis in Section IV-B and Section IV-C. Finally, we supplement the disucssion in Section IV-D.

A. Experimental Methodology

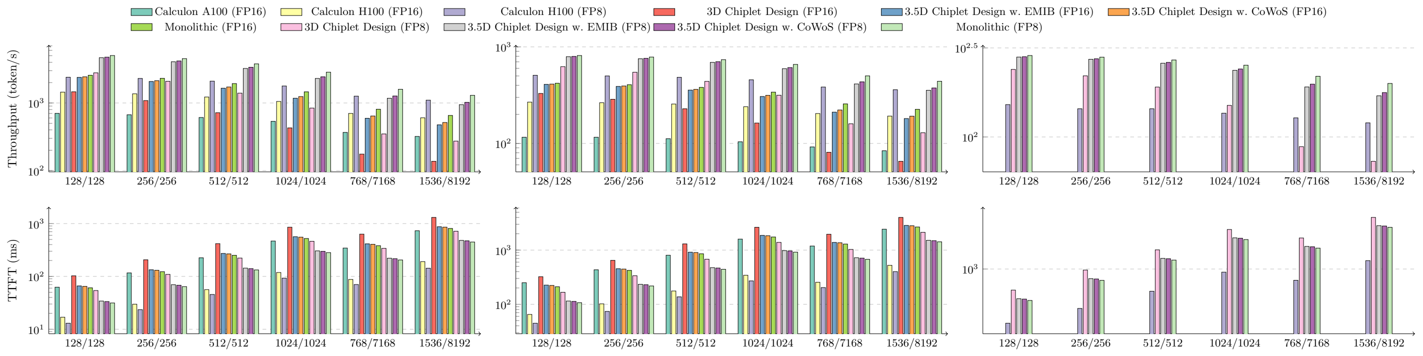
We detail the implementations, LLM workloads, and baselines as follows.

1) Implementations

We implement AccelStack in over 13K lines of Python code based on Calculon [34]. We derive architectural parameters, calibrate, and validate AccelStack using in-house data. Taking the monolithic design as an example (Fig. 1), the silicon area of the compute die is 32 × 25 mm^2, while the DRAM die maintains near-identical dimensions to ensure area matching. With four DRAM dies stacked, the total memory capacity reaches 64 GB. The design incorporates 16 PEs interconnected with a proprietary NoC. It delivers 786 TFLOPS of computing power at FP8 precision, and 393 TFLOPS at FP16 precision. Additionally, the 3D hybrid-bonded DRAM provides an aggregate bandwidth of 9.6 TB/s, while the NoC achieves a bisection bandwidth of 1.5 TB/s at the target frequency. For the chiplet design, we adopt UCIe as the implementation for chiplet-to-chiplet communication [23]. The 3.5D chiplet design with CoWoS (Fig. 4(b)) can achieve a bisection bandwidth of 1.1 TB/s, with edge latency ranging from 2 to 5 ns. The edge latency includes the latency of the adapter and the physical layer on TX/RX. The 3.5D chiplet design with EMIB achieves 1.0 TB/s bisection bandwidth. In contrast, the 3D chiplet design (Fig. 4(a)) delivers a bisection bandwidth of 255.0 GB/s. The peer-to-peer bandwidth for scaling-up systems is 800 GB/s. The coefficients used for cost analysis, *e.g.*, wafer price, α and β from Equation (5), *etc.*, are sourced from ICKnowledge [35]. The bonding price ratios for DoD, DoW, and WoW are set to $4:2:1$, and the hybrid bonding yield is set to 0.95 [36] [4]. We evaluate the cost of 3D-stacked LLM accelerators with a logic die fabricated in TSMC N5 (15 metal layers) and a DRAM die implemented in TSMC 16-nm technology. The peak power is below 400W, which can be dissipated by air cooling [27].

[3] It is worth noting that some interposers can be made from organics, so the number of these interposers can be approximated by $N_{\text{organic interposer}} = A_{\text{panel}}/A_{\text{organic interposer}}$, where A_{panel} is the rectangular raw organic material for producing these interposers.

[4] 0.95 is a conservative value, and the real yield can be higher than this value [37], [38].



Llama3 8B, 1 chip

Llama3 70B, 4 chips (TP = 4)

Llama3 405B, 8 chips (TP = 4, PP = 2)

Fig. 8 Performance comparison between Calculon A100, H100 and 3D-stacked LLM accelerator designs with specific LLM parallelization [34]. The top row of bar graphs shows throughput on a logarithmic scale, while the bottom row displays TTFT latency, also on a log scale. The notation 768/7168 denotes input/output sequence lengths of 768 and 7168 tokens respectively.

2) LLM Workloads

The LLMs evaluated are listed in TABLE I, covering a spectrum from popular Llama-series models to cutting-edge open-source workloads like DeepSeek-v3 [1], [39]. These workloads are different in scale and model architectural design, offering a thorough analysis with 3D-stacked LLM accelerator designs. The evaluation uses production traces, with input sequences of 128 $\sim$ 1536 tokens and output sequences ranging from 128 to 8192 tokens, representing LLM usage patterns from brief interactions to complex, long chain-of-thought reasoning tasks [2], [3], [40]. The batch size is 8, a number chosen in a typical production LLM service systems [40].

3) Baselines

We employ Calculon as the baseline for two reasons [34]. First, Calculon provides native support for Llama-series models and has been rigorously validated on NVIDIA's Selene supercomputer [26]. Second, since AccelStack itself builds upon Calculon, this choice ensures a fair and consistent comparison basis. We use Calculon to evaluate on both NVIDIA A100 and H100 platforms. For the A100, we conduct FP16 simulations as A100 lacks FP8 inference support, while for the H100, we evaluate at FP16 and FP8 precision [5]. To maintain model accuracy, we implement non-linear operations (including RoPE and softmax) in FP32 precision [41], [42]. Notably, the 3D-stacked LLM accelerator achieves 39.42% of the H100's FP8 compute performance (1979 TFLOPS) [6]. This margin-inclusive estimate stems from conservative silicon area allocation to the tensor cores, rather than by limitations in 3D integration.

B. Performance Evaluations

Section IV-B1 discusses the comparison with the baseline, while Section IV-B2 presents additional results.

1) Compare w. Baseline

We compare the monolithic design and the chiplet-based designs (with MCM, CoWoS, and EMIB, respectively) to the Calculon A100 and H100 via different precisions. Fig. 8 visualizes the results. Since Llama3 405B inference with FP16 precision exceeds available memory capacity, these results are omitted from Fig. 8.

At FP8 precision, the monolithic design, 3.5D chiplet design with CoWoS, and the 3D chiplet design achieve 1.86×, 1.80×, and 1.44× higher throughput, respectively, compared to the H100 for small outputs (*e.g.*, 128 tokens) with all tested Llama3 models. When evaluated at FP16 precision against the A100, the monolithic design shows 3.34× higher throughput, followed by the 3.5D chiplet designs with CoWoS (3.11×) and EMIB (3.03×), while the 3D chiplet design maintains 1.78× advantage. For long-sequence generation (7 $\sim$ 8K tokens) under test-time compute scaling [2], [3], the 3D-stacked LLM accelerator delivers 1.36× greater throughput than the H100 at FP8 precision. Compared to short sequences, the performance gains are reduced because Calculon does not account for the significantly increased KV cache size to a large degree in the context of long sequences. However, 3D-stacked LLM accelerators show limitations in time-to-first-token (TTFT) performance, with the monolithic design exhibiting an average of 2.33× slower TTFT than the H100 at FP8. This performance characteristic roots from TTFT's dependence on raw computing power, where the monolithic design's limited FP8 compute capability (as mentioned in Section IV-A3) constrains its efficiency. We also display the results compared to H100 at FP16.

2) Compare w. More LLM Workloads

Fig. 9 compares the performance of different 3D-stacked LLM accelerator designs at FP8 across more LLM workloads. For short generations, the monolithic design exhibits an average of narrow performance gap of 7.71% compared to the 3.5D chiplet design with CoWoS. However, for long-sequence generations, the monolithic design outperforms by 19.19%. Additionally, the monolithic design surpasses the EMIB design by 24.73% in the same case. In contrast, the 3D chiplet design suffers from limited chiplet-to-chiplet bandwidth due to MCM, which is 3.0× smaller than that of the monolithic design.

We can summarize key findings from Section IV-B1 and Section IV-B2. *First, 3D-stacked LLM accelerator can obtain higher throughput via enhanced memory bandwidth.* The memory bandwidth of the 3D-stacked LLM accelerator is approximately 3 $\sim$ 5× greater than H100. Even though the compute power of the 3D-stacked LLM accelerator is

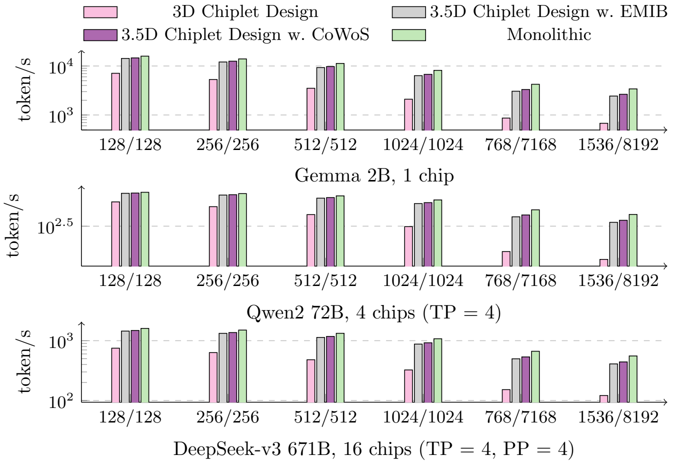


Fig. 9 Performance comparison between three 3D-stacked LLM accelerator designs with more LLM workloads at FP8.

substantially lower than that of the H100, it can still deliver up to 2.09× higher throughput at FP8 precision for Llama3 (7.17× over A100 at FP16), especially in scenarios involving shorter interactions with the LLM. *Second, chiplet-to-chiplet bandwidth can significantly impact performance, particularly during the generation of long sequences.* For short token generation, the performance gap between the monolithic and 3.5D chiplet designs with CoWoS is within 3.14%, but widens to 15.15% for long sequences. This suggests that the implementation of matrix multiplication should optimize chiplet-to-chiplet communication.

C. *Performance-per-Cost Evaluations*

We use performance-per-cost (throughput/cost) as a metric to evaluate the cost-effectiveness of each 3D-stacked LLM accelerator design. These designs utilize the DoD, DoW, or WoW manufacturing flows. The corresponding results are shown in Fig. 10. The X-axis represents shipment volume of chips, while the cost includes both NRE and RE costs, calculated using Equation (4). The performance values are derived from the Llama3 and DeepSeek-v3 model with different input and output sequence settings. For the Llama3 model, the 3.5D chiplet design with EMIB using the WoW manufacturing flow achieves the highest performance-per-cost for shipment volumes below 140K (*i.e.*, approximately 3500 logic wafers). However, when shipment volumes exceed this threshold, the monolithic design manufactured with DoD outperforms other designs. In contrast, the evaluation results differ for the DeepSeek-v3 model in long reasoning tasks. For shipment volumes below 30K (around 750 logic wafers), the 3.5D chiplet design with EMIB using the WoW manufacturing flow again achieves the highest performance-per-cost. For larger shipment volumes, nontheless, the monolithic design with DoD becomes more cost-effective, surpassing other designs by at least 17.32%. Although the 3D chiplet design reduces RE costs by over 38.09%, its limited chiplet-to-chiplet bandwidth leads to the lowest performance, resulting in the worst cost-efficiency among all designs.

A lesson can be learnt from the evaluations. Bandwidth should be the primary architectural consideration for larger models, particularly those requiring strong LLM reasoning capabilities. For shorter interactions with moderately sized LLMs, the 3.5D chiplet design with EMIB emerges as the optimal architectural solution.

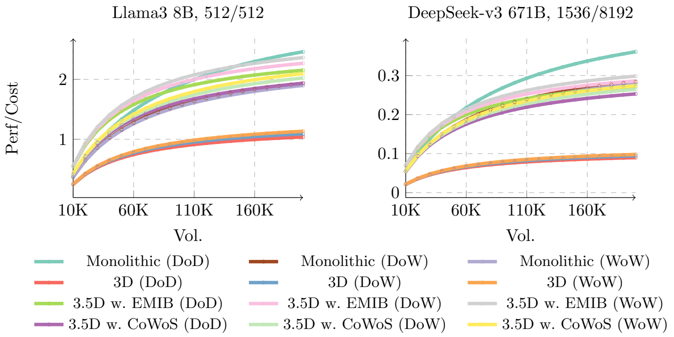


Fig. 10 Performance-per-cost analysis of 3D-stacked LLM accelerator designs for DeepSeek-v3.

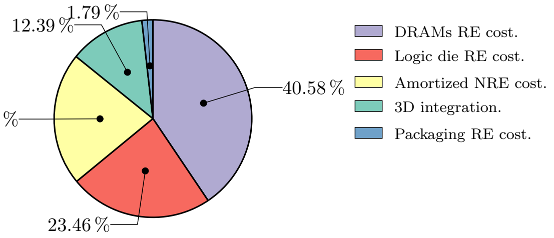


Fig. 11 Cost breakdown for monolithic design with WoW.

D. *Detailed Cost Analysis*

Fig. 11 illustrates the cost distribution for the monolithic design with WoW under 200K shipment volumes (Fig. 10). The majority of the cost is attributed to 3D hybrid-bonded DRAM (40.58%), followed by the logic die cost (23.46%). The 3D integration accounts for 12.39% of the total cost. In contrast, for the 3.5D chiplet design with CoWoS, packaging costs can reach up to 24.14% (not shown in Fig. 11), primarily due to the high expense of the silicon interposer.

V. Conclusions

In this paper, we present AccelStack, a cost-driven analysis framework for 3D-stacked LLM accelerators. Experimental results reveal that 3D-stacked LLM accelerators, despite utilizing only 39.42% of the computing power of H100's FP8, achieve an average throughput up to 2.03× and 7.17× higher than NVIDIA H100 and A100 simulations, respectively. In the performance-per-cost evaluations, as shipment volumes exceed 140K, the monolithic design with DoD demonstrates the highest cost-efficiency, followed by the 3.5D chiplet design with EMIB. Furthermore, 3D hybrid-bonded DRAM contributes 40.58% to the total cost of the monolithic design with WoW.

Acknowledgment

This research was partially supported by AI Chip Center for Emerging Smart Systems (ACCESS, sponsored by InnoHK funding), and Research Grants Council of Hong Kong SAR (16213824, 16212825, 16218324).

References

[1] A. Liu, B. Feng, B. Xue, B. Wang, B. Wu, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan *et al.*, "DeepSeek-V3 Technical Report," *arXiv preprint arXiv:2412.19437*, 2024.

[2] D. Guo, D. Yang *et al.*, "DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning," *arXiv preprint arXiv:2501.12948*, 2025.

[3] OpenAI, "OpenAI GPT-5 System Card," *https://cdn.openai.com/pdf/8124a3ce-ab78-4f06-96eb-49ea29ffb52f/gpt5-system-card-aug7.pdf*, 2025.

[4] H. Liao, J. Tu *et al.*, "Ascend: a Scalable and Unified Architecture for Ubiquitous Deep Neural Network Computing: Industry track paper," in *IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2021, pp. 789–801.

[5] J. Choquette, W. Gandhi, O. Giroux, N. Stam, and R. Krashinsky, "NVIDIA A100 Tensor Core GPU: Performance and Innovation," *IEEE Micro*, vol. 41, no. 2, pp. 29–35, 2021.

[6] J. Choquette, "NVIDIA Hopper H100 GPU: Scaling Performance," *IEEE Micro*, vol. 43, no. 3, pp. 9–17, 2023.

[7] N. P. Jouppi, G. Kurian *et al.*, "TPU v4: An Optically Reconfigurable Supercomputer for Machine Learning with Hardware Support for Embeddings," *IEEE/ACM International Symposium on Computer Architecture (ISCA)*, 2023.

[8] S. Lie, "Cerebras Architecture Deep Dive: First Look Inside the Hardware/Software Co-Design for Deep Learning," *IEEE Micro*, vol. 43, no. 3, pp. 18–30, 2023.

[9] A. Smith, G. H. Loh *et al.*, "Realizing the AMD Exascale Heterogeneous Processor Vision: Industry Product," in *IEEE/ACM International Symposium on Computer Architecture (ISCA)*. IEEE, 2024, pp. 876–889.

[10] "JEDEC Standard: High Bandwidth Memory (HBM3) DRAM, JEDEC Solid State Technology Association," 2023, https://www.jedec.org/standards-documents/docs/jesd238a.

[11] P. Huang, C. Lu, W. Wei, C. Chiu, K. Ting, C. Hu, C. Tsai, S. Hou, W. Chiou, C. Wang *et al.*, "Wafer Level System Integration of the Fifth Generation CoWoS®-S with High Performance Si Interposer at 2500 mm2," in *IEEE Electronic Components and Technology Conference (ECTC)*. IEEE, 2021, pp. 101–104.

[12] R. Mahajan, R. Sankman *et al.*, "Embedded Multi-Die Interconnect Bridge (EMIB) – A High Density, High Bandwidth Packaging Interconnect," in *IEEE Electronic Components and Technology Conference (ECTC)*. IEEE, 2016, pp. 557–565.

[13] R. Mahajan and S. Sane, "Advanced Packaging Technologies for Heterogeneous Integration," in *2021 IEEE Hot Chips 27 Symposium (HCS)*, 2021, pp. 22–24.

[14] M.-F. Chen, F.-C. Chen, W.-C. Chiou, and C. Doug, "System on Integrated Chips (SoIC (TM)) for 3D Heterogeneous Integration," in *IEEE Electronic Components and Technology Conference (ECTC)*. IEEE, 2019, pp. 594–599.

[15] K. Sakuma, R. Yu *et al.*, "D2W and W2W Hybrid Bonding System with Below 2.5 Micron Pitch for 3D Chiplet AI Applications," in *IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2024, pp. 1–4.

[16] Y.-M. Chen, T. Ko *et al.*, "Next Generation TSMC-SoIC® Platform for Ultra-High Bandwidth HPC Application," in *IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2024, pp. 1–4.

[17] R. Pope, S. Douglas, A. Chowdhery, J. Devlin, J. Bradbury, J. Heek, K. Xiao, S. Agrawal, and J. Dean, "Efficiently Scaling Transformer Inference," *Machine Learning and Systems (MLSys)*, vol. 5, pp. 606–624, 2023.

[18] W. Gomes, A. Koker, P. Stover, D. Ingerly, S. Siers, S. Venkataraman, C. Pelto, T. Shah, A. Rao, F. O'Mahony *et al.*, "Ponte Vecchio: A Multi-Tile 3D Stacked Processor for Exascale Computing," in *IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 65. IEEE, 2022, pp. 42–44.

[19] "Intel Foveros Direct," 2024, https://www.intel.com/content/dam/www/central-libraries/us/en/documents/2024-02/intel-tech-clearwater-wp.pdf.

[20] J. H. Lau, "Recent Advances and Trends in Advanced Packaging," *IEEE Transactions on Components, Packaging and Manufacturing Technology (TCPMT)*, vol. 12, no. 2, pp. 228–252, 2022.

[21] J. Xia, C. Cheng, X. Zhou, Y. Hu, and P. Chun, "Kunpeng 920: The First 7-nm Chiplet-Based 64-Core ARM SoC for Cloud Services," *IEEE Micro*, vol. 41, no. 5, pp. 67–75, 2021.

[22] S. Naffziger, N. Beck, T. Burd, K. Lepak, G. H. Loh, M. Subramony, and S. White, "Pioneering Chiplet Technology and Design for the AMD EPYC™ and Ryzen™ Processor Families: Industrial Product," in *IEEE/ACM International Symposium on Computer Architecture (ISCA)*. IEEE, 2021, pp. 57–70.

[23] "Universal Chiplet Interconnect Express 2.0 Specification," 2024, https://www.uciexpress.org/specifications.

[24] R. H. Bruce, W. P. Meuli, and J. Ho, "Multi Chip Modules," in *ACM/IEEE Design Automation Conference (DAC)*. IEEE Computer Society, 1989, pp. 389–393.

[25] Y. Feng and K. Ma, "Chiplet Actuary: A Quantitative Cost Model and Multi-Chiplet Architecture Exploration," in *ACM/IEEE Design Automation Conference (DAC)*, 2022, pp. 121–126.

[26] V. A. Korthikanti, J. Casper, S. Lym, L. McAfee, M. Andersch, M. Shoeybi, and B. Catanzaro, "Reducing Activation Recomputation in Large Transformer Models," *Machine Learning and Systems (MLSys)*, vol. 5, pp. 341–353, 2023.

[27] D. Niu, S. Li, Y. Wang, W. Han, Z. Zhang, Y. Guan, T. Guan, F. Sun, F. Xue, L. Duan *et al.*, "184QPS/W 64Mb/mm2 3D Logic-to-DRAM Hybrid Bonding with Process-Near-Memory Engine for Recommendation System," in *IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 65. IEEE, 2022, pp. 1–3.

[28] S. Wang, B. Yu, W. Xiao, F. Bai, X. Long, L. Bai, X. Jia, F. Zuo, J. Tan, Y. Guo *et al.*, "A 135 GBps/Gbit 0.66 pJ/bit Stacked Embedded DRAM with Multilayer Arrays by Fine Pitch Hybrid Bonding and Mini-TSV," in *2023 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*. IEEE, 2023, pp. 1–2.

[29] A. Alexandrov, M. F. Ionescu, K. E. Schauser, and C. Scheiman, "LogGP: Incorporating Long Messages into the LogP Model—One Step Closer Towards a Realistic Model for Parallel Computation," in *Proceedings of the seventh annual ACM symposium on Parallel algorithms and architectures (SPAA)*, 1995, pp. 95–105.

[30] "NVIDIA GB200 NVL72," 2025, https://nvdam.widen.net/s/wwnsxrhm2w/blackwell-datasheet-3384703.

[31] P. Zuo, H. Lin, J. Deng, N. Zou, X. Yang, Y. Diao, W. Gao, K. Xu, Z. Chen, S. Lu *et al.*, "Serving Large Language Models on Huawei CloudMatrix384," *arXiv preprint arXiv:2506.12708*, 2025.

[32] C. Wang, M. Chen, Q. Wang, Y. Fang, and L. Qiu, "New Product Development Paradigm from the Perspective of Consumer Innovation: A Case Study of Huawei's Integrated Product Development," *Journal of Innovation & Knowledge*, vol. 9, no. 2, p. 100482, 2024.

[33] D. Stow, I. Akgun, R. Barnes, P. Gu, and Y. Xie, "Cost Analysis and Cost-Driven IP Reuse Methodology for SoC design Based on 2.5D/3D Integration," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2016, pp. 1–6.

[34] M. Isaev, N. McDonald, L. Dennison, and R. Vuduc, "Calculon: a Methodology and Tool for High-Level Codesign of Systems and Large Language Models," in *IEEE/ACM International Conference for High Performance Computing, Networking, Storage, and Analysis (SC)*, 2023, pp. 1–14.

[35] "IC Knowledge LLC. IC Cost and Price Model; Assembly and Test Cost and Price Model." 2021, https://www.techinsights.com/press-release/techinsights-acquires-ic-knowledge-llc.

[36] A. P. Lujan, "Cost and Yield Analysis of Die-to-Wafer Hybrid Bonding," in *International Conference on Electronics Packaging (ICEP)*. IEEE, 2022, pp. 129–130.

[37] S. C. Chong, I. C. Daniel, S. L. P. Siang, J. S. C. Yi, A. L. W. Song, and W. L. Loh, "Yield Improvement in Chip to Wafer Hybrid Bonding," in *IEEE Electronic Components and Technology Conference (ECTC)*. IEEE, 2022, pp. 1982–1986.

[38] X. D. Chen, G. H. See, Y. W. Lim, P. Lianto, P. Suo, C. B. Y. Andy, S. K. Rath, and X. Zhao, "Integration Solution for Thin D2W Hybrid Bonding for Yield and Reliability," in *IEEE Electronic Components and Technology Conference (ECTC)*. IEEE, 2025, pp. 33–36.

[39] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and Efficient Foundation Language Models," *arXiv preprint arXiv:2302.13971*, 2023.

[40] H. Liu, C. Bai, P. Xu, L. Yin, X. Yu, H.-L. Zhen, M. Yuan, T.-Y. Ho, and B. Yu, "LLMShare: Optimizing LLM Inference Serving with Hardware Architecture Exploration," in *ACM/IEEE Design Automation Conference (DAC)*, 2025.

[41] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu, "Roformer: Enhanced Transformer with Rotary Position Embedding," *Neurocomputing*, vol. 568, p. 127063, 2024.

[42] L. Zou, W. Zhao, S. Yin, C. Bai, Q. Sun, and B. Yu, "BiE: Bi-Exponent Block Floating-Point for Large Language Models Quantization," in *International Conference on Machine Learning (ICML)*, 2024.